FTP Reference Manual

Table of Contents

Preface

Scope: The FTP Reference Manual describes in detail the user commands, software responses, and error codes for the FTP (File Transfer Protocol) file-transfer utility, including locally enhanced parallel file-transfer features. FTP is the standard interactive tool for moving files between machines or to (or from) the LC archival storage system (open or secure). LC's anonymous FTP service is also summarized here (open network only), as is the special encrypted-transfer tool called SFTP.

For alternative file-transfer tools that also rely on FTP software daemons to move files (but that offer special services beyond the basic FTP interface), see the <u>NFT Reference Manual</u> (URL: http://www.llnl.gov/LCdocs/nft) and the <u>HTAR Reference Manual</u> (URL: http://www.llnl.gov/LCdocs/htar) (both NFT and HTAR favor file transfers to or from storage, but they both have options that can enable more general transfers among other LC machines). For a clear comparison of FTP's features with those of NFT and SCP, and for a concise, task-oriented summary of how to use FTP for ordinary file transfers, see the <u>EZOUTPUT Basic Guide.</u> (URL: http://www.llnl.gov/LCdocs/ezoutput) For a comparison of FTP with alternative interfaces to LC's archival storage system, see the <u>EZSTORAGE Basic Guide.</u> (URL: http://www.llnl.gov/LCdocs/ezstorage)

Availability: FTP runs on all LC machines, open and secure. Fetch is an FTP client for Macintosh computers. Note that different vendors implement FTP details (such as listing suboptions) differently on different platforms, and system administrators can set FTP defaults (such as ascii or binary transfer mode) differently in different environments.

Consultant: For help contact the LC customer service and support hotline at 925-422-4531 (open e-mail: lc-hotline@llnl.gov, SCF e-mail: lc-hotline@pop.llnl.gov).

Printing: The print file for this document can be found at

```
OCF: http://www.llnl.gov/LCdocs/ftp/ftp.pdf
SCF: https://lc.llnl.gov/LCdocs/ftp/ftp_scf.pdf
```

Introduction

FTP (File Transfer Protocol) is an industry-standard protocol and user interface for transferring files between computer systems by means of a series of interactive commands. FTP involves a local client (software you execute to send or receive files) and a remote server (software elsewhere that responds to instructions from your client to accept or deliver files). FTP features:

- Use of standard TCP/IP network protocols to move files between machines.

- Support for transfers to or from nonUNIX systems as well as among computers runing UNIX.

- Use of IP host addresses (e.g., 134.9.55.221) as well as domain names (e.g., gps1.llnl.gov) to specify transfer targets.

- Interactive login, usually with password, to begin transfers to or from each remote machine (at LC, some transfers are preauthenticated to omit the password).

LC users with special file-transfer needs (such as for batch-oriented command files, extensive tracking of each transfer, or persistent transfers if network problems arise) may prefer to use the NFT (URL: http://www.llnl.gov/LCdocs/nft) file-transfer tool to move files among LC machines. See the EZOUTPUT Basic Guide (URL: http://www.llnl.gov/LCdocs/ezoutput) for a more elaborate comparison of FTP and NFT. Users whose primary interest in FTP is as an interface to LC's High Performance Storage System (archival file storage) may want to consult the EZSTORAGE Basic Guide (URL: http://www.llnl.gov/LCdocs/ezstorage) for helpful comparisons and alternative interfaces (such as LSTORAGE, CHMODSTG, CHGRPSTG, and HTAR). NFT but *not* FTP can manipulate "access control lists" (ACLs) on files stored in HPSS. If FTP transfer rates and reliability are important concerns for you (because you move large files, for example), then you can monitor recent FTP performance between many pairs of network nodes by using LC's NETMON web site (see the NETMON Reference Manual (URL: http://www.llnl.gov/LCdocs/netmon) for details).

This manual tells how to run FTP, shows a typical FTP session, and describes in detail all standard FTP interactive commands (options). Standard FTP server replies and error codes are included too, along with a brief introduction to LC's anonymous FTP service. On all LC production machines (but not necessarily on LC's other machines), a parallel FTP client (PFTP) is now the default, although actual parallel transfers sometimes must be overtly requested. Instructions for invoking a nondefault nonparallel FTP client are therefore also included (page 6) here, along with advice for handling the (sometimes trivial, sometimes complex) interaction between requested parallel transfers and jumbo-frame gigabit Ethernet links. Users who need to transfer files (to FIS) with their data encrypted can try SFTP, (page 34) a special FTP client with very limited server support.

How to Run FTP

Basic Execution

EXECUTE LINE.
To run FTP on any LC machine, type

      ftp [*remotehost*]

where *remotehost* is either the IP address (e.g., 134.9.55.221) or the domain name (e.g., gps1.llnl.gov) of the computer with which you want to exchange files. The machine on which you run FTP is the "client" or "local" machine, and the machine whose address or name you specify on the execute line is the "server" or "remote" machine (for purposes of describing commands and file transfers below). If run with no *remotehost*, FTP prompts for input (and you will need to use its interactive OPEN command to specify a target host).

You must log in to your local machine to run FTP, and you must also log in to the specified remote machine at the start of each FTP file-transfer session (when you are prompted for your remote user name and password, which might be different from the local ones). FTP expects file transfers to be done by a series of interactive commands, and it does not allow "third party" transfers (between two remote machines). To adapt FTP for use in batch scripts (by means of UNIX "here files"), see the annotated batch script in the EZJOBCONTROL Basic Guide. (URL: http://www.llnl.gov/LCdocs/ezjob)

FIREWALL ALERT.
LC now uses its hardware/software security "firewall" to block direct FTP connections from machines outside the llnl.gov domain to LC machines within llnl.gov. Offsite users must either (1) log on to some llnl.gov production machine, execute FTP there, and then draw external files toward them (with the GET option), or (2) become authorized VPN users and "borrow" an llnl.gov IP address with VPN before beginning their FTP session. See EZACCESS (URL: http://www.llnl.gov/LCdocs/ezaccess) or the Firewall and SSH Guide (URL: http://www.llnl.gov/LCdocs/firewall) for detailed instructions.

PARALLEL, NONPARALLEL CLIENTS.
On all LC production machines, open and secure (but not necessarily on LC's other machines), a parallel FTP client (PFTP) is now the default. When parallel file transfers actually occur depends on the pair of machines involved in the transfer (see next subsection). But the extra verbosity of the parallel client might sometimes pose problems (e.g., for scripts). To overtly invoke a "standard," NONparallel FTP client either type the special execute line

      ftp.bsd [*remotehost*]

or else set the ennvironment variable OLD_FTP_CLIENT to any nonnull value, for example

```
        setenv OLD_FTP_CLIENT 1
        export OLD_FTP_CLIENT=1
```

and then use the regular execute line shown at the start of this section (unless you include this setting in your login file, FTP will revert to the parallel default client with every new login session).

MONITORING FTP TRAFFIC.

NETMON, LC's network-monitoring web site, tracks FTP traffic (PUTs and GETs, for example, to storage.llnl.gov) from the perspective of several benchmark OCF and SCF network nodes. NETMON tracks each HPSS Class of Service (COS) separately. In fact, NETMON's tables and graphs *always* report FTP traffic by using HPSS Class of Service categories for file size even when storage.llnl.gov is not one of the nodes generating that FTP traffic (see the NETMON COS "slang" terminology in the right-most column of the chart below):

```
HPSS                                          NETMON
COS ID    Default for the file size    Slang term
------    --------------------------   ----------
 110      0 .GE. file .LT. 4 Mbyte      small (sFTP)
 120      4 .GE. file .LT. 32 Mbyte     medium (mFTP)
 130      32 .GE. file .LT. 256 Mbyte   large (lFTP)
 140      256 .GE. file                 jumbo (jFTP)
 150      [by request only]
```

The NETMON Reference Manual (URL: http://www.llnl.gov/LCdocs/netmon) tells how to interpret and customize NETMON's FTP-monitoring reports and plots.

SECURE FTP.

For information on how executing the specialized "secure FTP" (or SFTP) client differs from running standard FTP, consult the SFTP section (page 34) below.

TRANSFERRING ARCHIVE (TAR) FILES.

Suppose you want to bundle a set of files (perhaps including directory trees) and transfer the resulting archive to another LC machine, but you lack enough disk space to run TAR locally and (hence) double your disk usage on the client machine before you invoke FTP for the transfer. LC's special HTAR utility, formerly a storage interface only, now offers an option (-F) with which you can open a parallel connection to *any* preauthenticated LC FTP server and transfer files directly into an archive (TAR file) created and filled only on the *remote* (receiving) machine. Extractions from and even indexing of remote nonstorage archives are also supported. See the HTAR Reference Manual (URL: http://www.llnl.gov/LCdocs/htar) for details and annotated examples. With HTAR, archive member files can be as large as 8 Gbyte, archives can contain up to 1,000,000 member files, and there is no limit on total archive size. (HTAR manages these file transfers internally, not by executing the PFTP client.)

Parallel FTP Service

Parallel FTP service is available between each LC production machine and (both OCF and SCF) storage, and between pairs of LC production machines themselves. In some cases it is automatic (automatic parallelization depends on the FTP daemon on the *target* machine, not on the client). In other cases you must request it. (HTAR automatically uses parallel transfers but does not execute the PFTP client.)

Also, some (OCF and SCF) LC machines are now connected to "jumbo-frame gigabit Ethernet links" for fast network traffic. Naturally, the best file-transfer rates occur when you invoke parallel FTP between a pair of machines that also has the fast jumbo-frame links (rates between 35 and 50 Mbyte/s are possible for single FTP sessions).

This table summarizes the available parallel/jumbo transfer combinations. Usage details and a more elaborate explanation of the conditions for automatic parallel transfers appear in the two subsections below.

```
FTP                 Kind of FTP connection
Destination     Parallel       Jumbo-frame
-----------     --------       -----------

Storage         automatic      yes (IBM, CHAOS to storage)
                (over 1 MB)    no (Compaq to storage)

AIX or          automatic      yes (among IBMs, SGIs)
OCF Tru64                      no (among Compaqs, Suns)

other           by request     yes (among IBMs, SGIs)
(Linux)                        no (among Compaqs, Suns)
```

Using Jumbo-Frame (High-Speed) Links

Network communication problems are traditionally subdivided and solved in "layers." The now-standard OSI ("open system interconnect") network model involves seven such layers, as follows:

```
Layer   Layer          Networking      Units
Number  Name           Role            Exchanged
------  ----           ----            ---------

7       application    supports user      |
6       presentation   programs and     messages
5       session        processes          |

4       transport      implements
3       network        protocol suite   packets

2       data link      specifies        frames
                       network features

1       physical       specifies        bits, bytes
                       hardware features
```

At the lowest ("physical") layer, mere electrical signals are exchanged on the network, representing bits and bytes. But just above that layer, the network exchanges something with a meaningful internal structure, a "frame." A frame consists of your data "encapsulated" (flanked) by headers and a trailer of extra bytes used for routing, tracking, and reliability:

```
-------------------------------------------------
| Ethernet | IP     | other   | your | trailer |
| header   | header | headers | data |         |
-------------------------------------------------
```

A standard Ethernet frame contains 1500 bytes of user data, plus its headers and trailer. A "jumbo" frame instead contains 9000 bytes of user data, so that the percentage of overhead for the headers and trailer is much less and data-transfer rates can be much higher.

In May, 2001, LC installed "jumbo-frame gigabit Ethernet links" on both OCF and SCF storage systems, and on some (ASCI) production machines. Use of these jumbo-frame network links for faster file transfers is automatic among those LC computers that have them installed. The default (parallel) FTP client on these production machines checks the environment variable PFTP_CONFIG_FILENAME, discovers the list of available jumbo-frame links in the reference file /etc/pftp_config, and automatically uses these links to improve file-transfer rates. Some machines have multiple jumbo-frame links, allowing multiple high-speed file-transfer sessions at once. (HTAR checks the file /usr/local/etc/HPSS.conf rather than PFTP_CONFIG_FILENAME to find parallel connections to storage.)

This table shows which LC machines (all IBM, Linux/CHAOS, or SGI) have deployed jumbo-frame gigabit Ethernet links, and it lists the number of links/machine. All other LC machines (mostly Compaqs and Suns) lack jumbo-frame links.

```
Machine               Jumbo-frame
Name                  Links
----                  ----

SCF storage           14 (4/single transfer)
White login nodes      4
Tidalwave              4
Riptide                4
Edgewater              4
Adelie login nodes     4
Emperor login nodes    4
MCR login nodes        4
OCF storage            3 (2/single transfer)
Frost                  2
SKY login nodes        1
Ice login nodes        1
```

Requesting Parallel Transfers

The default FTP client now on all LC production machines (but not necessarily on special-purpose machines) is a locally developed version that enables you to transfer data in parallel. Some FTP parallel transfers occur automatically, while some you must request.

AUTOMATIC.
The FTP server ("daemon") on your *destination* (target) machine determines whether or not the file transfer is automatically parallel. (Only) machines with the WU LLNL-24 daemon version (or later) support automatic parallelization.

(A) For all files over 1 Mbyte, FTP file transfers to storage from all LC production machines (both directions) are automatically parallel (OCF and SCF). Transfers originating on those machines with "jumbo-frame gigabit Ethernet links" (see previous subsection (page 9)) also automatically use those links for even faster data movement. Storage transfers are the easiest parallel case; no special commands are needed. (All HTAR transfers to stored archive files are also automatically parallel, but they do not involve the PFTP client.)

(B) File transfers to any LC AIX (IBM) and to any OCF Tru64 (Compaq) node from all LC production machines are also automatically parallel. After the "user logged in" message at the start of each FTP session, the server on these target machines explicitly announces "Server supports parallel features. Auto-parallel substitution enabled." On these machines (only) FTP's PARALLEL command is effectively disabled: typing PARALLEL reports the current stripe width and block size, but it does *not* enable or disable subsequent parallel file transfers.

ON DEMAND.
FTP file transfers to other (than the above) LC production machines (including to Linux nodes) are parallel only on request. These target machines lack the automatic-parallel daemon and hence no "auto-parallel" message greets you at the start of each FTP session. Typing the PARALLEL command will enable parallel transfers here (details below). However, after the first use further uses of PARALLEL merely report the stripe width and block size but do *not* disable subsequent parallel transfers (until your FTP session ends). Because LC's parallel FTP client interacts with jumbo-frame network links if there are any, you must also take account of their presence or absence when you overtly request parallel transfers. There are two subcases, depending on where you run the FTP client:

(A) WITH JUMBO-FRAME LINKS: If you run FTP on a machine with jumbo-frame links (see the list in the previous subsection (page 9)) and exchange files (in either direction) with another machine that also has jumbo-frame links, then either accept automatic parallel transfers (if the target is an AIX or storage node) or invoke parallel file transfer simply by typing the command

```
parallel
```

in response to any ftp> prompt before you use GET or PUT. If you want to run FTP on a machine with jumbo-frame links to exchange files (either direction) with a machine that LACKS jumbo-frame links (e.g., IBM to Compaq transfer), then *before* executing FTP type

```
unset PFTP_CONFIG_FILENAME
```

to prevent the FTP client from fruitlessly trying to find a jumbo-frame path to or from the target machine. When you subsequently run FTP, type its

```
parallel
```

command as before to overtly request parallel (but nonjumbo) transfers. In either case, the PARALLEL command reports the stripe width and block size of the parallel transfers that it enables.

(B) WITHOUT JUMBO-FRAME LINKS: If you run FTP on a machine without jumbo-frame links (any machine NOT listed in the previous subsection (page 9)), then there is no problematic interaction with the PFTP_CONFIG_FILENAME environment variable regardless of what you choose for a target machine. If the target is *not* AIX or storage, you can invoke parallel transfers during any FTP session to any other target machine simply by typing the command

```
parallel
```

in response to any ftp> prompt before you use GET or PUT.

PFTP COMMANDS.
The PFTP client offers nine extra commands (beyond the usual set offered by FTP) to specifically manage parallel file transfers (for example, PGET and MPGET perform parallel GETs). On LC production machines these special PFTP commands are quite unnecessary because using the local PARALLEL command (above) enables all the same benefits with other standard FTP commands (GET, etc.) without your having to learn any new syntax. At other (ASCI tri-lab) sites, where PARALLEL is not implemented, you may need to remember the special PFTP commands to perform parallel file transfers (especially to storage). See LC's HPSS User Guide (URL: http://www.llnl.gov/LCdocs/hpss) for details on the nine extra PFTP commands.

OPTIONAL EXTRAS.
LC's parallel FTP client is more verbose than the standard FTP client during file transfers. Parallel FTP users may want a complete record of each verbose FTP dialog in their batch log files. The undocumented execute-line option -c causes all interactive output to be sent during batch runs of FTP as well. The undocumented -e option copies FTP input commands into your batch output. Thus running FTP with the execute line

```
ftp -ce remotehost
```

will preserve all the details of a parallel session even within a batch job.

FTP sessions with storage (and with some other target machines) are fully preauthenticated and do not prompt for your *username*, while in other cases FTP returns a

```
Name (host:username):
```

prompt to which you must reply to continue. Parallel users who want to eliminate this Name: prompt from all sessions (including batch sessions) can install a file called .netrc in their (global) home directory, containing the THREE lines

```
default
```

```
login username
```

where last line in the .netrc file must be present but blank.

Sample FTP Session

The following sample session (with annotated steps added along the right side) shows a typical dialog by which a user (JANE) transfers files interactively using FTP. In this case, the local machine (on which Jane executes the FTP client) is GPS1, and the remote machine that files are copied to and from is LUCY.

```
(1) The user runs FTP (on GPS1) with the remote machine's domain
    name as an argument.
(2) FTP prompts for a userid and a password to log in to LUCY
   (some LC machines "preauthenticate" and skip this password step).
(3) At the ftp> prompt, the user changes remote directories
    to /var/tmp/jane (which is not shared among LC machines).
(4) At the next ftp> prompt, the user GETs file NFT.PS
    (copies it from LUCY to GPS1).
(5) At the next ftp> prompt, the user PUTs file TESTFILE
    (copies it from GPS1 to LUCY).
(6) The user then requests parallel file transfers, and transfers
    a 1.8 Mbyte file called LARGE from GPS1 to LUCY using 4 parallel
    stripes (each separately reported as FTP runs).
(7) When the file transfers are done and confirmed, the user
    QUITs FTP.
```

```
ftp lucy.llnl.gov                                           ---(1)

Connected to lucy.llnl.gov.
220 [NOTICE TO USERS -- very long legal statement]
222 lucy.llnl.gov FTP server (Version LLNL-22...) ready.
202 Command not implemented.
Name (lucy.llnl.gov:jane): jane                             ---(2)
331 Password required for jane.
Password: [does not echo]
230 User jane logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
Multinode is Disabled.

ftp> cd /var/tmp/jane                                       ---(3)
250 CWD command successful.

ftp> get nft.ps                                             ---(4)
200 PORT command successful.
150 Opening Binary data connection for nft.ps
226 Binary Transfer complete.
1602470 bytes received in 0.579 seconds (2.64 Mbytes/s)

ftp> put testfile                                          ---(5)
200 PORT command successful.
150 Opening Binnary data connection for testfile
226 Transfer complete.
5264 bytes sent in 0 seconds (5.14 Kbytes/s)


ftp> parallel                                              ---(6)
Parallel Features and Auto-Parallel Substitution Enabled.
Parallel block size set to (1048576).
```

```
Parallel stripe width set to (4).

ftp> put large
200 Command complete (1827811, large, 0)
200 Command complete.  Address 1 is 134.9.55.221.10100
200 Command complete.  Address 2 is 134.9.55.221.10101
200 Command complete.  Address 3 is 134.9.55.221.10102
200 Command complete.  Address 4 is 134.9.55.221.10103
150 Transfer starting.
226 Transfer complete. (moved = 1827811).
1827811 bytes sent in 0.20 seconds (8.95 Mbytes/s)
200 Command complete.

ftp> quit                                               ---(7)
221 Goodbye.
```

Standard FTP Commands

This section lists frequently-used standard FTP commands, their definitions, usage, and examples. On many systems, you may enter the bracketed portion of the command as a shortcut. The examples use the shortcut version of the commands. Commands may only be entered at the FTP prompt, not on FTP's execute line.

Additional FTP commands might be supported by your local FTP client, and if so, they might be described in your local FTP MAN pages. Additional helpful FTP commands may also be supported by some FTP servers (such as those on LC's archival storage system). To pass an unsuported local command (such as CHMOD) to a remote FTP server that does support it, you must use FTP's QUOTE command (page 24), described below.

[Ap]pend

Appends a local file to a file on the remote machine. If the remote file is left unspecified, the local file name is used.

Usage: append *local-file remote-file*
Example: ap /usr/bob/newfile /users/comp/bob/file1

[As]cii

Sets the file transfer type to ASCII. This is commonly the default.

Usage: ascii
Example: as

Binary files should never be transfered while in ASCII mode, otherwise they will be transfered incorrectly. ASCII mode performs character translations of certain characters. These characters occur randomly in a binary file and their translation may result in the corruption of the binary file.

[Bi]nary

Sets the file transfer type to support binary image transfer. Binary file transfer is faster than ASCII file transfer, since character and record translations are not performed. (Binary is the default for transfers to or from LC's archival storage system, open and secure.)

Usage: binary
Example: bi

[By]e

Terminates the FTP session with the remote server and exits FTP. BYE is a synonym for QUIT.

Usage: bye

Example: by

Cd

Changes the current directory on the remote machine to the directory specified. To change the local directory (on the client machine), use the LCD (page 19) command instead.

Usage: cd *remote-dir*
Example: cd /users/comp/bob

[Cdu]p

Changes the current remote-machine directory to its parent directory.

Usage: cdup
Example: cdu

[Cl]ose

Terminates the current FTP session with the remote server and returns to the command interpreter (the FTP prompt), but does not terminate FTP. Once you CLOSE the connection, you may use the OPEN (page 22) command to connect to another remote server.

Usage: close
Example: cl

[Del]ete

Deletes the specified remote file from a remote directory. (To delete files and directories recursively, use NFT (URL: http://www.llnl.gov/LCdocs/nft) instead of FTP.)

Usage: delete *remote-file*
Example 1: del file3
Example 2: del /users/comp/bob/dir1/file1

Dir

Lists the entries (files) and entry attributes of a directory for the remote machine. If the remote directory is not specified, the current remote directory will be listed. DIR often accepts display-control suboptions, but they vary from one FTP implementation to another. See also LS (page 20). (To list files and directories recursively, use NFT (URL: http://www.llnl.gov/LCdocs/nft) instead of FTP.)

Usage: dir [*remote-directory*]
Example: dir /users/comp/bob

[Ge]t

Copies a remote file from a remote directory to the local machine. If no pathnames are specified, FTP will copy the *remote-file* from the current remote directory to the current local machine directory.

Usage: get *remote-file* [*local-file*]
Example 1: ge /users/comp/bob/file1 /usr/bob/newfile
Here /users/comp/bob/file1 is copied from the remote machine to /usr/bob/newfile on the local machine.

Example 2: ge file1 newfile
Here file1 is copied from the current remote directory to newfile in the current local machine directory.

Example 3: ge file1 /usr/bob/newfile
Here file1 is copied from the current remote directory to /usr/bob/newfile on the local machine.

Example 4: ge /users/comp/bob/file1
Here /users/comp/bob/file1 is copied from the remote machine to /users/comp/bob/file1 on the local machine if the directory /users/comp/bob exists. If /users/comp/bob does not exist on the local machine, the command will return an error.

NOTE:
If you want to GET a few files from inside a remote TAR-format archive file *without* first GETting the whole (large) archive to your local machine, use HTAR (with -F) instead of FTP. Consult the HTAR Reference Manual (URL: http://www.llnl.gov/LCdocs/htar) for instructions. You can selectively monitor FTP GET traffic by using NETMON's (URL: http://www.llnl.gov/LCdocs/netmon) Report Builder feature. (To GET files and directories recursively, use NFT (URL: http://www.llnl.gov/LCdocs/nft) instead of FTP.)

[Ha]sh

Toggles the printing of the hash (or pound) sign (#) on the screen for each data block transferred. The size of the data block is machine dependent.

Usage: hash
Example: ha

[He]lp

Displays information to help you learn more about FTP local client commands. There are two ways to access **help**. If you type the command **help** by itself, a list of valid FTP commands will display. To get help on a specific command, type **help** followed by the command name. Use REMOTEHELP (page 25) to get information on FTP server, rather than client, commands.

Usage 1:   help

Example 1: he

Usage 2:   help *command-name*
Example 2: he get

Lcd

Changes the current directory on the local machine. If no directory is specified, FTP changes to your local home directory. (To change directories on the remote (server) machine, use the <u>CD</u> (page 17) command instead.)

Usage: lcd [*local-directory*]
Example: lcd /usr/bob

Ls

Lists file names in a remote directory. If the remote directory is not specified, FTP lists your current remote directory. Use DIR (page 17) to list files and their attributes. To list stored files, even recursively for all levels of your storage directories, you may find the separate LSTORAGE tool more effective than LS or DIR (see EZSTORAGE (URL: http://www.llnl.gov/LCdocs/ezstorage) for instructions).

Usage: ls [*remote-directory*]
Example: ls /users/comp/bob

[Mde]lete

Deletes multiple files. This command deletes the specified remote files. If full pathnames are not specified, the default is the current remote directory. Wildcard characters (*, ?, [ ]) may be used with **mdelete**. To disable the interactive prompting that occurs by default for each file when you use **mdelete**, use the **prompt** command.

Usage: mdelete *remote-file*1 *remote-file*2 ...
Example 1: mde file1 file2
Here file1 and file2 are deleted from the current remote directory.

Example 2: mde /users/comp/bob/file1 /users/comp/barb/file1
Here /users/comp/bob/file1 and /users/comp/barb/file1 are deleted from the specified directories.

Example 3: mde test*
Here your local FTP client expands the file filter test* into a file list, but each remote server may process that expanded filter differently. At LC, the HPSS (storage) server for FTP interprets **mdelete** recursively, and it removes all (matching) files not only in the current remote directory (in storage) but also in the directory children of that directory as well. See the "FTP Pitfalls (with Storage)" section of EZSTORAGE (URL: http://www.llnl.gov/LCdocs/ezstorage) for several ways to handle this aggressive interpretation of **mdelete** when your remote FTP server is HPSS.

[Mge]t

Gets multiple files. This command lets you retrieve specified files from one or more remote directories and transfer them to the current local directory. Wildcard characters (*, ?, [ ]) may be used with **mget**. To disable the interactive prompting that occurs by default for each file when you use **mget**, use the **prompt** command.

Usage: mget *remote-file*1 *remote-file*2 ...
Example 1: mge file1 file2
Example 2: mget /users/comp/bob/file1 /users/comp/barb/file1

[Mk]dir

Makes a directory on the remote machine. If no pathname is specified, the directory will be inserted into the current remote directory.

Usage: mkdir *remote-dir*
Example: mk /users/comp/bob/dir2

[Mp]ut

Puts multiple files. Transfers multiple local files from one or more local directories to the current remote directory. Wildcard characters (*, ?, [ ]) may be used with **mput**. To disable the interactive prompting that occurs by default for each file when you use **mput**, use the **prompt** command.

Usage: mput *local-file*1 *local-file*2 *local-file*3 ...
Example: mp file1 file2 file3
Here three files are transferred from the current local directory to the current remote directory.

[Op]en

Establishes a connection to the specified host (remote) FTP server. If an optional port number is specified, FTP will attempt to contact an FTP server at that port. If you did not specify a target host when executing FTP, you must so do with the OPEN command before you can transfer files. See also CLOSE (page 17).

   Usage: open host [*port*]
   Example 1: op remote_machine
   Example 2: op remote_machine 1021

[Par]allel

Enables parallel file transfers (LC production machines only). Parallel transfers are ON by default whenever you transfer a file over 1 Mbyte to or from storage or *to* any AIX or OCF Tru64 node. But parallel transfers are OFF by default between any other pair of production machines, including Linux nodes (so typing **parallel** once here enables them).

   Usage: parallel
   Example: par
Here the FTP client responds with "Auto-parallel substitution enabled," and it reports the current parallel block size and the current parallel stripe width. Once enabled, parallel transfers persist until your FTP session ends. See the Parallel FTP Service (page 8) section for more details.

[Prom]pt

Toggles interactive prompting. This occurs during multiple file transfers to allow the user to retrieve and store files selectively. If prompting is turned off, an **mget**, **mput**, or **mdelete** command operates on all files specified without prompting for your approval. The default prompt value is on.

   Usage: prompt
   Example: prom

[Pu]t

Copies a local file from a local directory to the remote machine. If no pathnames are specified, PUT copies the *local-file* from the current local directory to the current remote machine directory.

Usage: put *local-file* [*remote-file*]
Example 1: pu /usr/bob/file1 /users/comp/bob/newfile
Here /usr/bob/file1 is copied from the local machine to /users/comp/bob/newfile on the remote machine.

Example 2: pu file1 newfile
Here file1 is copied from the current local directory to newfile in the current remote machine directory.

Example 3: pu file1 /users/comp/bob/newfile
Here file1 is copied from the current local directory to /users/comp/bob/newfile on the remote machine.

Example 4: pu /usr/bob/file1
Here /usr/bob/file1 is copied from the local machine to /usr/bob/file1 on the remote machine if the directory /usr/bob exists on the remote machine. If /user/bob does not exist on the remote machine, the command returns an error.

NOTE:
If you want to PUT a large TAR-format archive file into storage (or to *any* LC machine with a preauthenticated FTP server) but don't have the space (or time) to build it first on your local machine, use HTAR (with -F) instead of FTP (HTAR will actually build the archive directly on the *remote* machine as member files arrive). Consult the HTAR Reference Manual (URL: http://www.llnl.gov/LCdocs/htar) for details. You can selectively monitor FTP PUT traffic by using NETMON's (URL: http://www.llnl.gov/LCdocs/netmon) Report Builder feature. (To PUT files and directories recursively, use NFT (URL: http://www.llnl.gov/LCdocs/nft) instead of FTP.)

[Pw]d

Prints the name of the current remote directory.

Usage: pwd
Example: pw

[Qui]t

Terminates the FTP session with the remote server and exits FTP. Same command as **bye**.

    Usage: quit
    Example: qui

[Quo]te

    Sends specified arguments or commands verbatim to the remote FTP server. Some FTP clients accept supplementary commands without a prefix (e.g., **chmod**), some clients expect **site** as a prefix for supplementary site-specific commands (such as commands to HPSS for file storage at LC), and some clients require the longer **quote site** prefix (e.g., **quote site stage**). Using the **quote site** long form is always the safest strategy to try. See Appendix C (URL: http://www.llnl.gov/LCdocs/hpss/index.jsp?show=apc) of the HPSS User Guide for a list of quoted commands currently accepted by the FTP server on LC's archival storage system (open and secure).

    Usage: quote *arg1* [*arg2*]...
    Example: quo site chmod 775 myfile
    Example: [quote] site setcos 150

    NOTE: if you run FTP to manage your stored files at LC, you may find it easier to use the separate tools CHMODSTG and CHGRPSTG to change permissions and storage groups, even recursively. See EZSTORAGE (URL: http://www.llnl.gov/LCdocs/ezstorage) for instructions. NFT (URL: http://www.llnl.gov/LCdocs/nft) can also change storage permissions and storage groups recursively.

Recv

    A synonym for **get**.

    Usage: recv *remote-file* [*local-file*]
    Example: See GET (page 18).

[Rem]otehelp

Lists the commands supported by the FTP server to which you have connected (but not in any useful order). Commands marked with * are unimplemented. Commands listed with quotes must be used by means of the **quote** command (but many available quoted commands may not be listed). [Rh]elp is a synonym for remotehelp. See <u>HELP</u> (page 18) for information client (rather than server) commands.

  Usage: remotehelp [*command*]
  Example: rem
  Example: rem stage

[Ren]ame

Renames the file *from-name* on the remote machine to the file *to-name*. Renaming a file is the way to move it between remote directories since no MV command is available with FTP.

  Usage: rename *from-name to-name*
  Example: ren /users/comp/bob/newfile /users/comp/bob/oldfile

[Res]et

Clears the reply queue. This command resynchronizes command/reply sequencing with the remote FTP server.

  Usage: reset
  Example: res

[Rmd]ir

Deletes a directory on the remote machine. If no pathname is specified, it deletes the directory from the current remote directory. You must empty a remote directory before you can delete it. (To delete directories recursively, without emptying them, use <u>NFT</u> (URL: http://www.llnl.gov/LCdocs/nft) instead of FTP.)

  Usage: rmdir *remote-directory*
  Example: rmd /users/comp/bob

[Rsta]tus

Displays the current status of the FTP server (includes the version number, your user ID, the client machine name, and current data transfer mode). If a file or directory is specified, attribute information is listed as well. "PFTPD" in the response confirms a parallel FTP server (such as on LC's storage machines). See <u>STATUS</u> (page 26) for client information.

Usage: rstatus [*remote-file*]
Example: rsta

Send

A synonym for **put** .

Usage: send *local-file* [*remote-file*]
Example: See PUT (page 23).

Site

See the section above (page 24) for **quote** .

[Sta]tus

Displays the current status of the FTP client (includes mode, verbosity, case, and hash settings, and proxy connections). STATUS on LC's production machines will confirm the presence of a parallel client (autoparallel on, parallel block size declared) by default. See RSTATUS (page 25) for server information.

Usage: status
Example: sta

[Us]er

    Identifies you to the remote FTP server as the same or a different user. If you enter only the login-name, you will be prompted for a password (if required). You may supply your login-name and password at the time the command is initially entered.

  Usage: user *login-name* [*password*]
  Example 1: user bob
User bob will be prompted to enter a password.

  Example 2: us bob XXXX
User bob has entered his login-name and his password already.

Common FTP Replies and Errors

When you enter an FTP command, you receive a corresponding reply that indicates that the command was accepted, rejected or is being processed. An FTP reply consists of a three-digit code followed by a brief description of the result. This section contains an overview of common FTP reply codes for the commands discussed in <u>Standard FTP Commands</u> (page 16). For a complete listing of FTP replies in numeric order, see <u>FTP Replies</u> (page 32).

append	• Accepted:
	  125 Using existing data connection for *filename*.
	  150 Opening *mode* data connection for *filename*.
	  226 Transfer complete.
	
	• Rejected:
	  421 Service not available, remote server has closed connection.

ascii	• Accepted: 200 Type set to A.
binary	• Accepted: 200 Type set to I.
bye, close, quit	• Accepted: 221 Goodbye (system status).

cd, cdup	• Accepted:
	  250 Remote CWD command successful.
	
	• Rejected:
	  421 Service not available, remote server has closed connection.
	  550 *path: error*.

delete, mdelete	• Accepted:
	  250 Remote DELE command successful.
	
	• Rejected:
	  421 Service not available, remote server has closed connection.
	  550 *filename:* No such file or directory.
	  550 *filename: error*.
	  550 *filename:* cannot delete .trash directory.

dir	• Accepted:
	  150 Opening ASCII mode data connection.
	  200 Port command successful.
	  226 Transfer complete.
	
	• Rejected:
	  500 Data connection: *error*.

get, mget, recv	• Accepted:
	  125 Using existing data connection for *remote-file*.
	  150 Opening *mode* mode data connection for *remote-file* (*size*).
	  200 Port command successful.
	  226 Transfer complete.
	
	• Processing:
	  550 File *remote-file* is being moved from the archive.
	
	• Rejected:
	  421 Service not available, remote server has closed connection.
	  426 Data connection: *error*.
	  550 *remote-file*: *error*.

hash	• Accepted:
	  Hash mark printing on (1024 bytes/hash mark).
	  Hash mark printing off.

help	• Accepted: Help text (command list).

lcd	• Accepted:
	  Local directory now *pathname*.
	
	• Rejected:
	  pathname: error.

ls	• Accepted:
	  150 Opening ASCII mode data connection.
	  200 Port command successful.
	  226 Transfer complete.
	
	• Rejected:
	  426 Service not available, remote server has closed connection.

mkdir	• Accepted: 257 Remote MKD command successful. • Rejected: 550 *remote-directory: error.*
open	• Accepted: 220 Remote FTP server ready. 230 User *username* logged in, proceed. • Rejected: 421 Service not available, remote server has closed connection.
prompt	• No reply code; replies whether interactive mode is on or off.
put, mput, send	• Accepted: 125 Using existing data connection for *local-file*. 150 Opening *mode* mode data connection for *local-file*. 200 Port command successful. 226 Transfer complete. • Rejected: 421 Service not available, remote server has closed connection. 550 *local-file*: No such file or directory.
pwd	• Accepted: 257 *pathname* is current directory.
remotehelp, rhelp	• Accepted: 214 Help text. • Rejected: 502 Unknown command *command*.
rename	• Accepted: 250 Remote RNTO command successful. 350 File exists, ready for destination name. • Rejected: 550 rename: *error.* 550 *from-name*: No such file or directory.
reset	• No reply code if accepted.

rmdir	• Accepted:
	250 Remote RMD command successful.
	• Rejected:
	550 *remote-directory*: No such file or directory.
	550 *remote-directory*: Not a directory.
	550 Cannot delete .trash directory.
rstatus	• Accepted:
	221 Server status information.
status	• Accepted:
	Current status of FTP shown.
	• Rejected:
	421 Service not available, remote server has closed connection.
user	• Accepted:
	230 User logged in, proceed.
	• Processing:
	331 Password required for *login-name*.
	• Rejected:
	530 Login incorrect.

FTP Reply Codes

Listed below are FTP reply codes (in numeric order) and their meanings. The exact text accompanying each reply code depends on the command issued. For some examples, see <u>Common FTP Replies and Errors</u> (page 28), above.

```
Reply     Meaning
Code

125       Data connection already open, transfer starting.
150       File status okay, about to open data connection.
200       Command successful.
202       Command not implemented, superfluous at this site.
213       File status.
214       Help message.
220       Service ready for new user.
221       Service closing control connection.
225       Data connection already open, no transfer in progress.
226       Closing data connection.
230       User logged in, proceed.
250       Requested action okay, completed.
257       Pathname is current directory.
          Command successful.
258       Command on (or off).
331       User name okay, need password.
350       Requested file action pending further information.
421       Service not available, closing control connection.
425       Can't open data connection.
426       Connection closed, transfer aborted.
451       Requested action aborted, local error in processing.
452       Requested action not taken.
500       Syntax error, command unrecognized.
501       Syntax error in parameters or agruments.
502       Command not implemented.
503       Bad sequence of commands.
504       Command not implemented for that parameter.
505       No such file or directory.
          File being moved from the archive.
506       Usage error.
522       Transfer error bytes written.
530       Error in user login.
550       Requested action not taken due to error.
551       Requested action aborted.
553       Requested action not taken due to system error.
          User not authorized to use command.
```

Anonymous FTP

To support the exchange of files (especially large or nontext files) among LLNL collaborators, LC provides anonymous FTP servers on both the open and secure networks. Anyone can contact these servers (by running FTP with the user name ANONYMOUS and any password) to exchange files placed there by LC colleagues. One directory allows anyone to PUT, another allows anyone to GET, and a third directory supports PUTs and GETs exclusively for those within the llnl.gov domain. The domain name of LC's open anonymous FTP server is

```
ftp.llnl.gov
```

and specific instructions for using it appear in the Anonymous FTP Service (URL: http://www.llnl.gov/LCdocs/ezoutput/index.jsp?show=s5.4.4) section of the EZOUTPUT Basic Guide.

LC's secure-network anonymous FTP server, formerly at

```
ftp.scf.cln
```

has been disabled for security reasons.

SFTP (Secure FTP)

ROLE:

Standard FTP clients do not encrypt the data that they send to remote hosts, which theoretically allows malicious third parties to intercept and read that data. Secure FTP (SFTP) is a modified client that does encrypt all the files that it sends for greater safety. However, only suitable SFTP servers can accept file transfers from SFTP clients (because SFTP uses SSH2 and talks to the SSHD2 daemon, not to the usual FTPD or WU-FTPD daemons).

AVAILABILITY:

- Clients--
  OCF:
  SFTP clients now reside on all OCF production machines (including AIX/IBM, Linux, and Tru64/Compaq unclassified machines).
  SCF:
  SFTP clients are *not* available (yet) on any LC secure-network computers.

- Servers--
  FIS:
  LC's File Interchange Service (FIS, at fis.llnl.gov) is the only LC server that now accepts incoming files from SFTP clients. And even FIS only accepts SFTP transfers from within the LC firewall, so direct SFTP transfers from "outside" machines by means of OTS, IPA, or VPN are *not* accepted.
  Others:
  No other LC FTP servers accept SFTP transfers. In particular, you can *not* store files (at storage.llnl.gov) from any host by running SFTP.

DIALOG DIFFERENCES:

SFTP clients present a different user dialog than do standard FTP clients on LC machines. While some differences are trivial, others require different user responses to open connections or to transfer files successfully. SFTP...

- Does *not* request your username (nor present it as a default to which you can simply respond with RETURN).

- Checks for a "host key" for every new host to which you try to connect and, if not found, asks if you want to continue connecting (yes/no) anyway.

- Requests your one-time password (OTP) to open every no-host-key connection (no default preauthentication occurs, unlike for standard FTP connections among LC machines).

- Displays the usual long security prolog at the start of each SFTP session, but often in a distorted, unreadable format.

- Prompts for input with SFTP>.

OPTION DIFFERENCES:

SFTP recognizes only 14 of the usual set of 35 <u>FTP control options.</u> (page 16) Among the most useful options that SFTP accepts are:

open, close

quit
cd, lcd
pwd
get, mget
put, mput
mkdir, rmdir
help [lists all SFTP options].

Among the most useful standard FTP options that SFTP does *not* accept are:

dir (to list files and their properties). In most situations, the SFTP alternative option
ls -l
reports the same information as DIR does for standard FTP sessions.

delete (to remove files). The SFTP alternatives
rm [removes remote files],
lrm [removes local files],
perform the same functions as DELETE does during standard FTP sessions.

ascii
binary
parallel
quote
site

for which SFTP provides *no* alternative options. SFTP software is supposed to automatically detect ASCII and BINARY files on arrival and transfer them in the appropriate mode, but you cannot force the mode if inappropriate transfers occur.

KEYS:
If you prefer not to use your OTP (one-time password) to authenticate every SFTP session, you can create and install a special file called an "SSH public DSA key," generated using either OpenSSH or F-Secure SSH2, on every pair of machines between which you transfer files with SFTP. Generating an appropriate DSA key, converting it to the needed OpenSSH format if needed, and installing it in the right directories (including those on the open FIS node) is a complex, multi-step process. For explicit instructions on how to set up SFTP for key authentication instead of password authentication (but no other SFTP details), retrieve and print LC's Technical Bulletin 308, available on both the open and secure networks by requesting this URL with your web browser (note the 's' in https):

https://lc.llnl.gov/computing/techbulletins/bulletin308.html

Disclaimer

Keyword Index

To see an alphabetical list of keywords for this document, consult the

```
Keyword                     Description
-------                     -----------
entire                      This entire document.
title                       The name of this document.
scope                       Topics covered in this document.
availability                Where these programs run.
who                         Who to contact for assistance.

introduction                FTP features introduced.

execute-line                How to run FTP.
   ftp-usage                Basic FTP client execution.
   parallel-usage           How to request parallel FTP transfers.
      jumbo-frames          Jumbo frames defined, inventoried.
      parallel-cases        How parallel and jumbo FTP interact.
   ftp-example              Typical FTP interactive session.

ftp-commands                FTP standard commands.
   append                   Append to a remote file.
   ascii                    ASCII (text) transfer mode.
   binary                   Binary (image) transfer mode.
   bye                      Terminate FTP.
   cd                       Change remote directories.
   cdup                     Move to remote parent directory.
   close                    Close FTP session without quitting.
   delete                   Delete remote file.
   dir                      List remote files and attributes.
   get                      Retrieve one remote file.
   hash                     Toggle hash-mark indicator.
   help                     List commands and descriptions.
   lcd                      Change local directories.
   ls                       List remote file names.
   mdelete                  Delete multiple remote files.
   mget                     Retrieve multiple remote files.
   mkdir                    Make remote directory.
   mput                     Transfer multiple local files.
   open                     Start a new FTP session.
   parallel                 Toggle parallel file transfers.
   prompt                   Toggle interactive prompting.
   put                      Transfer one local file.
   pwd                      Reveal current remote directory.
   quit                     Terminate FTP.
   quote                    Pass commands through to server.
   recv                     Retrieve one remote file.
   remotehelp               List remote server commands.
   rename                   Rename (or move) remote file.
   reset                    Clear FTP reply queue.
   rmdir                    Delete remote directory.
   rstatus                  Show FTP server status.
   send                     Transfer one local file.
   site                     Pass commands through to server.
   status                   Show FTP client status.
```

```
  user                    Transmit user name and password.

ftp-errors                FTP reply and error codes expained.
ftp-replies               FTP reply codes by number.
anonymous-ftp             LC's anonymous FTP servers.

sftp                      Secure FTP role, features, limits.

index                     The structural index of keywords.
a                         The alphabetical index of keywords.
date                      The latest changes to this document.
revisions                 The complete revision history.
```

Alphabetical List of Keywords

```
Keyword                     Description
-------                     -----------

a                           The alphabetical index of keywords.
anonymous-ftp               LC's anonymous FTP servers.
append                      Append to a remote file.
ascii                       ASCII (text) transfer mode.
availability                Where these programs run.
binary                      Binary (image) transfer mode.
bye                         Terminate FTP.
cd                          Change remote directories.
cdup                        Move to remote parent directory.
close                       Close FTP session without quitting.
date                        The latest changes to this document.
delete                      Delete remote file.
dir                         List remote files and attributes.
entire                      This entire document.
execute-line                How to run FTP.
ftp-commands                FTP standard commands.
ftp-errors                  FTP reply and error codes expained.
ftp-example                 Typical FTP interactive session.
ftp-replies                 FTP reply codes by number.
ftp-usage                   Basic FTP client execution.
get                         Retrieve one remote file.
hash                        Toggle hash-mark indicator.
help                        List commands and descriptions.
index                       The structural index of keywords.
introduction                FTP features introduced.
jumbo-frames                Jumbo frames defined, inventoried.
lcd                         Change local directories.
ls                          List remote file names.
mdelete                     Delete multiple remote files.
mget                        Retrieve multiple remote files.
mkdir                       Make remote directory.
mput                        Transfer multiple local files.
open                        Start a new FTP session.
parallel                    Toggle parallel file transfers.
parallel-cases              How parallel and jumbo FTP interact.
parallel-usage              How to request parallel FTP transfers.
prompt                      Toggle interactive prompting.
put                         Transfer one local file.
pwd                         Reveal current remote directory.
quit                        Terminate FTP.
quote                       Pass commands through to server.
recv                        Retrieve one remote file.
remotehelp                  List remote server commands.
rename                      Rename (or move) remote file.
reset                       Clear FTP reply queue.
revisions                   The complete revision history.
rmdir                       Delete remote directory.
rstatus                     Show FTP server status.
scope                       Topics covered in this document.
send                        Transfer one local file.
sftp                        Secure FTP role, features, limits.
site                        Pass commands through to server.
```

<u>status</u>	Show FTP client status.
<u>title</u>	The name of this document.
<u>user</u>	Transmit user name and password.
<u>who</u>	Who to contact for assistance.

Date and Revisions

```
Revision    Keyword       Description of
Date        Affected      Change
--------    --------      ------
01Sep04     ftp-usage     HTAR features updated.
            parallel-usage Details updated.
            jumbo-frames  Linux availability added.


09Feb04     introduction  Use NFT to handle remote ACLs.
            ftp-commands  Recursive NFT alternatives noted.


03Dec03     ftp-usage     HTAR size limits clarified.
            scope         NFT, HTAR can do nonstorage transfers.


28Jul03     ftp-usage     HTAR now also to/from NONstorage hosts.
            get           HTAR for nonstorage archive gets too.
            put           HTAR for nonstorage archive puts too.


28Apr03     parallel-usage More automatic parallel transfers explained.
            parallel      Usage revised for new servers.


04Feb03     sftp          New section added on secure FTP.
            index         New keyword for new section.
            introduction  SFTP role noted.
            ftp-usage     Cross reference to SFTP added.


14Nov02     ftp-usage     NETMON FTP monitoring by file size.
            get           NETMON GET monitoring available.
            put           NETMON PUT monitoring available.


09Sep02     anonymous-ftp Anonymous SCF service disabled.


19Jun02     mdelete       Warning added on storage use.
            quote         SETCOS example added.
            site          New section added.
            index         New keyword for new section.


04Feb02     introduction  NETMON cross reference added.
            ftp-usage     WEST replaced in example.
            ftp-example   Example dialog updated.


28Aug01     scope         HTAR role added.
            introduction  HTAR role added.
            get           When to use HTAR instead of GET.
            put           When to use HTAR instead of PUT.


11Jul01     parallel-cases PFTP intro, cross ref added.


26Jun01     parallel-usage New section explains parallel transfers.
            parallel      New (LC) FTP command.
            ftp-example   PARALLEL command illustrated.
            index         New keywords for new subsections.


22Mar01     introduction  Default parallel clients noted.
            ftp-usage     How to run nonparallel client added.
            ftp-example   Parallel client more verbose.
```

```
               status          Parallel client clues noted.
               rstatus         Parallel server clues noted.

     08Jun00   scope           Print instructions added.
               introduction    EZSTORAGE role explained.
               ftp-usage       Access restrictions updated.
               quote           CHMODSTG, CHGRPSTG alternatives noted.
               ls              LSTORAGE alternative noted.
               anonymous-ftp   New directories compared.

     01Apr99   execute-line    Firewall blocks FTP now.

     03Feb99   scope           SCP replaces RCP.
               introduction    Anonymous FTP service noted.
               ftp-usage       Firewall alert added.
               anonymous-ftp   Drop-off service deleted.

     06Aug98   entire          First edition of this document.


     TRG (01Sep04)
```